DECENTRALIZED MULTI-USER MULTI-ARMED BANDITS WITH USER DEPENDENT REWARD DISTRIBUTIONS

BY

AKSHAYAA MAGESH

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Electrical and Computer Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2020

Urbana, Illinois

Adviser:

Professor Venugopal V. Veeravalli

ABSTRACT

The uncoordinated spectrum access problem is studied using a multi-player multi-armed bandits framework. We consider a decentralized multi-player stochastic multi-armed bandit model where the players cannot communicate with each other and can observe only their own actions and rewards. Furthermore, the environment may appear differently to different players, *i.e.*, the reward distributions for a given arm may vary across players. Knowledge of time horizon T is *not* assumed. Under these conditions, we consider two settings - zero and non-zero reward on collision (when more than one player plays the same arm). Under the zero reward on collision setting, we present a policy that achieves expected regret of $O(\log T)$ over a time horizon of duration T. While settings with non-zero rewards on collisions and varying reward distributions of arms across players have been considered separately in prior work, a model allowing for both has not been studied previously to the best of our knowledge. With this setup, we present a policy that achieves expected regret of order $O(\log^{2+\delta} T)$ for some $0 < \delta < 1$ over a time horizon of duration T.

To my family, for their love and support.

ACKNOWLEDGMENTS

I wish to express my sincere gratitude to my advisor, Prof. Venugopal V. Veeravalli, Henry Magnuski Professor of Electrical and Computer Engineering at the University of Illinois Urbana-Champaign, for his constant guidance throughout the course of my master's degree. Without his persistent help, this work would not have been possible.

I wish to express my deepest gratitude to my parents and my sister for their unconditional support and patience.

I would also like to thank my friends Aditya and Rupesh for their constructive discussions and continuous encouragement.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

The multi-armed bandit is a well-studied framework to model sequential decision-making problems with an inherent exploration-exploitation trade-off. Multi-armed bandits have applications in recommendation systems, advertising, ranking results of search engines, and more. The classical stochastic multi-armed bandit setup considers an agent/player, who at each time instant t, chooses an action from a finite set of actions (or arms). The agent receives a reward drawn from an unknown distribution associated with the arm chosen. The goal is to come up with a decision-making policy that maximizes the agent's cumulative reward or equivalently minimizes regret. Policies that are designed to minimize regret in bandit settings aim to achieve sub-linear regret with respect to the time horizon T. The multi-armed bandit problem was first considered in the context of clinical trials by Thompson [1], who introduced a posterior sampling heuristic commonly known as Thompson sampling. In their seminal work, Lai and Robbins [2] formalized the stochastic multi-armed bandit setting and provided a lower bound on regret of order $\Omega(\log T)$ for time horizon T. They also presented an asymptotically optimal decision policy that introduced the idea of upper confidence bounds, which was further explored by [3] and [4]. Other variants of the multi-armed bandit setup such as adversarial, contextual and Markovian have also been studied in literature [5, 6].

Recently, there has been a growing interest in the study of *multi-player* multi-armed bandit settings, where instead of a single agent, there are K agents simultaneously pulling the arms. Depending on the level of coordination between players, various policies have been proposed to cooperatively find the best arm or to maximize the sum of rewards of all players. Multi-player bandit setups where agents cooperate have applications in geographically distributed ad servers [7], peer-to-peer networks [8], and distributed recommendation systems [9]. Cesa-Bianchi et al. [7] study a multi-player

adversarial bandit setting and [9] considers a multi-player contextual linear bandit setting.

Multi-player stochastic multi-armed bandit settings are particularly relevant to cognitive radio and dynamic spectrum access systems. In these systems, the finite number of channels representing different frequency bands are treated as arms, the users in the network are treated as the players, and the data rates received from the channels can be interpreted as the rewards. Since the maximum rate that can be received from a channel is limited, assuming bounded rewards for the arms is justified. Current spectrum management protocols treat frequency spectrum as a fixed commodity, which leads to spectrum underutilization. Dynamic spectrum access techniques have emerged as good strategies to improve spectrum utilization. Existing techniques for dynamic spectrum access have focused primarily on the primary/secondary user paradigm, where secondary users detect vacant bandwidths when available and vacate the occupied channel when a primary user wants to transmit. In this work, we consider the uncoordinated spectrum access model, where there is no such hierarchy among users and users are not allowed to communicate with each other. In this setting, players are competing for the same set of finite resources, and the expected regret is defined with respect to the optimal assignment of arms that maximizes the sum of expected rewards of all players (which can be interpreted as the system performance). The event of multiple players pulling the same arm simultaneously is commonly referred to as a *collision* and leads to players receiving reduced or zero rewards. Thus, while designing policies for the multi-player setting, in addition to balancing the exploration-exploitation trade-off, it is important to control the number of collisions that occur.

In the presence of a central controller in a multi-player system, the problem reduces to a single agent multi-armed bandit setup where the agent can choose multiple arms at a time. However, due to the communication overhead placed by a central controller, it is important to study a decentralized system. A tight (in the order sense) lower bound for the system regret for the centralized case is of course the same as that for a single agent multi-armed bandit setup, *i.e.*, $\Omega(\log T)$, which also serves as a lower bound on system regret for the decentralized case. It should be noted that no larger lower bounds have been proven for the decentralized case.

Some of the prior work in the decentralized setting assumes that commu-

nication between players is possible [10, 11]. However cooperation through communication between players imposes an additional cost and may suffer from latency issues due to delays. Other works assume that sensing occurs at the level of every individual player, such as smart devices in a network being able to sense if a channel is being used or not without transmitting on it. The work in [12] considers such a setting where a distributed auction algorithm is used by players to come to a consensus on the optimal assignment of arms. However in other systems, such as emerging architectures in Internet of Things (IoT), the individual nodes are not capable of sensing. This motivates the study of a completely distributed scenario, where there is no central control and the players cannot communicate with each other in any manner. The players can observe only their own actions and rewards.

In the fully distributed setting with no communication, in most of prior work in literature, the assumption is made that the reward distribution for any arm is the same across all players (homogeneous setting). This setting was first considered in [13], where prior knowledge of number of players is not assumed. The algorithm presented, named Multi-user ϵ-Greedy collision Avoiding algorithm (MEGA), combines the probabilistic ϵ-greedy algorithm with a collision avoiding mechanism inspired by the ALOHA protocol, and provides guarantees of sub-linear regret. Rosenski et al. [14] proposed an algorithm named Musical Chairs (MC), that is composed of a learning phase for the players to learn an ϵ-correct ranking of arms and number of players and a 'Musical Chairs' phase, in which the K players fix on the top K arms. They provide high probability guarantees of constant regret. The fully distributed setting with homogeneous reward distributions across players is also considered in [15, 16, 17]. All the above mentioned works also assume that in the event of a collision, all the colliding players receive zero rewards.

In cognitive radio and uncoordinated spectrum access networks, users are usually not colocated physically, and thus the reward distributions for a given arm may vary across users (heterogeneous setting). There have been few works that study the heterogeneous setting. Kalathil et al. [12] and Tibrewal et al. [18] consider such a setup, although they assume that the players are capable of sensing (*i.e.*, players can observe whether an arm is being used or not without pulling it). However, such an assumption might be unrealistic for the uncoordinated spectrum access problem.

In the first part of this thesis, we consider a fully distributed setting where

players can observe only their own actions and rewards, and allow for heterogeneous reward distributions. In the event of a collision, colliding players get zero rewards. The work in [19] considers such a fully distributed setting, and the proposed algorithm achieves a regret of $O(\log^2 T)$. The work in [16] introduces the idea of using forced collisions as a way to communicate among the users in the homogeneous setting where the reward distributions for the channels are the same across users. In this part of our work, we use the idea of forced collisions in the heterogeneous case and present a policy that achieves expected regret of $O(\log T)$. The algorithm presented here was developed independently of the work in [20], where an approach similar to ours is explored, *i.e.*, a fully distributed setting with user dependent rewards and with zero reward on collision. However, while the algorithm presented in [20] achieves logarithmic regret only in the case of a unique optimal matching, our policy results in logarithmic regret even in the case of multiple optimal matchings.

Another assumption that needs to be closely examined is that of zero rewards on collisions. In practical scenarios, when more than one player transmits on a channel in an uncoordinated wireless network, the colliding players may receive reduced, and not necessarily zero, rates or rewards. Thus allowing for non-zero rewards on collisions results in a more realistic model. Bande and Veeravalli [21] have considered a setting with homogeneous reward distributions across players and non-zero rewards on collisions. They also allow for the number of players to be greater than the number of arms. The algorithm presented in [21] is an extension of the Musical Chairs algorithm by [14] and provides high probability guarantees of constant regret. In the second part of this thesis, we study a multi-player multi-armed bandit setup which allows for heterogeneous reward distributions and non-zero rewards on collisions. We also allow for the number of players to be greater than the number of arms. Our work is based on [22] and, in contrast to the work in [19], requires modifications to results in [22] to accommodate non-zero rewards on collisions. To the best of our knowledge, ours is the first work to consider a model that allows for both heterogeneous reward distributions and non-zero rewards on collisions. In this setting, we propose an algorithm that achieves sub-linear regret of $O(\log^{2+\delta} T)$ for a time horizon T, with $0 < \delta < 1$.

CHAPTER 2

ZERO REWARD ON COLLISION

2.1 System Model

We consider the scenario of a multi-user game involving K users and M channels as the arms in a stochastic multi-armed bandit setup. We assume that $K < M$ and that the rewards for the channels are bounded in $[0, 1]$. Let the mean reward for user j on channel m be denoted by $\mu_j(m)$. Consider a time horizon T, and let the action taken by user (arm chosen by the user) j at time $t \leq T$ be $a_{t,j}$. In the case of a collision, *i.e.*, when multiple users access the same channel, all the colliding users receive zero reward.

Let $\mathcal{A}(K, M)$ denote all the possible user channel matchings, *i.e.*, $\mathbf{a} = [a_1, a_2, ..., a_K] \in \mathcal{A}(K, M)$, with a_j denoting the action taken by user j. Since we assume that colliding users get zero rewards, we only consider matchings that are unique, *i.e.*, once for which all the users are assigned to distinct arms. Let $\mathbf{a}^* \in \mathcal{A}(K, M)$ be such that

$$\mathbf{a}^* \in \underset{\mathbf{a} \in \mathcal{A}(K,M)}{\arg\max} \sum_{j=1}^{K} \mu_j(a_j).$$

Define J_1 to be the system reward for the optimal matching, *i.e.*, $J_1 = \sum_{j=1}^{K} \mu_j(a_j^*)$, and J_2 to be the system reward for the second optimal matching. In our algorithm, we assume that we have access to a lower bound on the parameter Δ defined as follows (see also [12], [19]):

$$\Delta = \frac{J_1 - J_2}{2M}.$$

Note that this quantity is strictly positive even in the case of multiple optimal matchings.

The expected regret of the system is defined as

$$R(T) = T \sum_{j=1}^{K} \mu_j(a_j^*) - \mathbb{E}\left[\sum_{t=1}^{T}\sum_{j=1}^{K} \mu_j(a_{t,j})\right]$$

where the expectation is over the actions of the players.

2.2 Algorithm

We assume that the players are time synchronized and that they enter the system at $t = 0$. The algorithm proceeds in epochs for each user. This allows us to proceed without knowing the time horizon T. Each epoch has three phases.

The first phase is the exploration phase, which has two parts. The first part is the fixing phase and is done for each user to obtain a unique ID. Each user accesses the arms uniformly and once a free arm is found, *i.e.*, non-zero reward is received, that arm is played for the rest of the fixing phase. The channel numbers they settle on serve as their IDs. At the end of the fixing phase, the users that are not fixed occupy channel 1, and the fixed users access channel 1 in order of their IDs sequentially. If the fixed users do not face a collision during this step, all users have obtained unique IDs. Once all the users obtain unique IDs, say at epoch ℓ_f, this part of the exploration phase is no longer done from epoch $\ell_f + 1$. The second part of the exploration phase is for the users to get estimates of the mean rewards of the arms. The users start from the channels corresponding to their IDs, and sample each arm for γ time units in a round-robin fashion.

The second phase is the matching phase and its purpose is for the users to arrive at the optimal matching. The first part of the matching phase is for each user to communicate their estimates of the mean rewards of all the channels to the other users. The users transmit the estimated mean rewards $\hat{\mu}_j(m)$ for all the channels in the order of their IDs. Since the players are not allowed to directly communicate with each other, collisions are used as a way to exchange information among players. The use of forced collisions as a form of communication was introduced in [16]. The main idea is that there are M channels available to the users and the transmitting user j

occupies a certain channel. When the receiving users access the channels one at a time, the channel number on which they face a collision gives them information about what was being transmitted. The value of the estimate $\hat{\mu}_j(m)$ that each user has received at the end of the matching phase is denoted by $\hat{\hat{\mu}}_j(m)$. Note that this is similar to truncating the value of $\hat{\mu}_j(m)$ to a finite number of bits as $\hat{\hat{\mu}}_j(m)$. At the end of the first part of the matching phase, $|\hat{\mu}_j(m) - \hat{\hat{\mu}}_j(m)| \leq \Delta/2$ for $j \in [K]$ and $m \in [M]$ given that all the users have a unique ID.

Once this communication part of the matching phase is completed, each user has the same approximated values of estimated mean rewards. Each user then independently solves for the set of optimal matchings from the matrix of $\hat{\hat{\mu}}_j(m)$ values. If there is a unique optimal matching, the users play the arm according to that matching for the exploitation phase. If there are multiple optimal matchings, it is necessary for each user to choose the same optimal matching from this set. This is achieved in the following manner. The user with ID number one chooses one of the optimal matchings and occupies that channel. The remaining users access the M channels in order of their IDs to get the channel number chosen by the user with ID one and update the set of optimal matchings that correspond with the channel chosen by user one. This process is repeated until all the users settle on the same optimal matching. This matching is played by all the users for the exploitation phase.

In our algorithm, the estimated mean rewards of all the channels are communicated to each user through the idea of forced collisions, and hence all the users can independently solve the assignment problem to arrive at the same set of optimal matchings. However, in [20], the communication mechanism is adapted from [16], where a leader-follower protocol is employed. The followers send the values of estimated mean rewards to the leader, and the leader computes the matching that has to be played by the users. While the algorithm presented here gives guarantees of logarithmic regret for the cases of unique optimal matching and multiple optimal matchings, the work in [20] provides guarantees of logarithmic regret only for unique optimal matchings and quasi-logarithmic regret in the case of multiple optimal matchings. Note that the constants in the upper bound for average regret of our algorithm are comparable to the ones obtained in [20].

Algorithm 1: Decentralized MUMAB Algorithm

Initialization: Set $\hat{\mu}_j(m) = 0$ for all values of $j \in [K]$ and all $m \in [M]$ and L_T as the last epoch with time horizon T.

for $\ell = 1, ..., L_T$ **do**

Exploration phase:

Fixing phase : Access channels uniformly for T_f time units till find a channel with no collision; fix on that for remainder of sub-phase. The channel number fixed on serves as unique ID.

If not fixed during fixing phase, send a flag by occupying channel 1.

If fixed during part fixing phase, access channel 1 in order of ID. Once all users are fixed, skip fixing phase.

Access each channel in a round robin fashion for γ time units to get estimates $\hat{\mu}_j(m)$.

Matching phase: Enter the matching algorithm to convey the estimates to all users, receive their estimates and calculate the optimal matching.

Exploitation phase: Occupy the channel resulting from the matching algorithm for 2^ℓ time units.

end

2.3 Regret Analysis

Theorem 1. *Assuming the rewards of each channel for all users are bounded in* $[0, 1]$ *and i.i.d. for all* $t \leq T$ *for a time horizon* T *and* $\Delta = \frac{J_1 - J_2}{2M}$ *is known, the regret of the decentralized MUMAB algorithm is* $O(\log T)$

Proof. The regret incurred during the L_T epochs can be analyzed as the sum of the regrets incurred in the three stages of the algorithm. From the structure of the epochs, we have that $L_T < \log T$.

1. Exploration phase: Let the regret incurred during the exploration phase for all epochs be R_1. The exploration goes on for $T_f + K + \gamma M$ time units till epoch ℓ_f (when all the users get fixed) and for γM time units after that. Choosing $T_f = M \log(20K)$ and $\gamma = \frac{1}{2\Delta^2}$, we have that

$$\begin{aligned} R_1 &\leq \sum_{\ell=1}^{L_T} \left(M \left(\frac{1}{2\Delta^2} + 1 + \log 20K \right) \right) \\ &\leq \left(M \left(\frac{1}{2\Delta^2} + 1 + \log 20K \right) \right) \log T. \end{aligned} \tag{2.1}$$

Algorithm 2: Matching Algorithm for user i

Initialization: Transmit in order of ID.

for *Transmitting user* $j' = 1, ..., K$ **do**

If turn to transmit, transmit $\hat{\mu}_{j'}(m)$ for $m = 1, ..., M$ as:

Occupy channel $h_1 = \lceil M\hat{\mu}_{j'}(m) \rceil$ for M time units.

Occupy $h_r = \lceil M^r(\hat{\mu}_{j'}(m) - \sum_{n=1}^{r-1} \frac{h_n - 1}{M^n}) \rceil$ for M time units for each subsequent round r for $\frac{1}{\log M} \log(\frac{1}{\Delta})$ rounds.

If not turn to transmit, in order of IDs:

Access channels in round robin fashion and initialize the estimates of the transmitted value as $\hat{\hat{\mu}}_{j'}(m) = \frac{2h-1}{2M}$ in the first round.

Update in the subsequent round r as $\hat{\hat{\mu}}_{j'}(m) = \sum_{n=1}^{r-1} \frac{h_n - 1}{M^n} + \frac{2h_r - 1}{2M^r}$.

end

Calculate the set S_M of optimal matchings with values of estimates.

if S_M *is a singleton set* **then**

Assign channel according to the optimal matching.

else

User 1 chooses one of the optimal matchings.

Remaining users update their set of optimal matchings accordingly.

Similar mechanism for subsequent users allows users to settle on one of the optimal matchings.

end

2. Matching phase: Let the regret incurred during the matching phase be R_2. The matching phase runs for $\frac{KM^3}{\log M}\log\frac{1}{\Delta}+(M-k)M^2+M^3$ when there are multiple optimal matchings and for $\frac{KM^3}{\log M}\log\frac{1}{\Delta}+(M-k)M^2$ time units when there is an unique optimal matching. Thus

$$\begin{aligned} R_2 &\leq \sum_{\ell=1}^{L_T}\frac{KM^3}{\log M}\log\frac{1}{\Delta}+(M-k)M^2+M^3 \\ &\leq \left(\frac{K}{\log M}\log\frac{1}{\Delta}+2\right)M^3\log T. \end{aligned} \tag{2.2}$$

3. Exploitation phase: Let R_3 denote the regret incurred during the exploitation phase. Regret is incurred in the exploitation phase in epoch ℓ only in case of the following two events:

   (a) The users do not get a unique ID in the first part of the exploration phase. Let $P_\ell(A)$ denote the probability that after the exploration phase of epoch ℓ the users do not have a unique ID

   (b) Given that users have unique IDs, for some $j\in[k]$ and some $m\in[M]$, $|\hat{\hat{\mu}}_j(m)-\mu_j(m)|>\Delta$. Let $P_\ell(B)$ denote the probability of this event.

   Thus we have that

$$R_3=\sum_{\ell=1}^{L_T}2^\ell(P_\ell(A)+P_\ell(B)). \tag{2.3}$$

   Let p_f denote the probability that with the fixing phase running for T_f time units, all users are fixed. From Lemma 1 of [16], we have that $p_f\geq(1-Ke^{\frac{T_f}{M}})$. From the definition of the event A, we have that

$$P_\ell(A)=(1-p_f)^\ell. \tag{2.4}$$

   Our choice of T_f results in $p_f>\frac{3(e-1)}{2e}$ and thus,

$$\sum_{\ell=1}^{L_T}2^\ell P_\ell(A)=\sum_{\ell=1}^{L_T}2^\ell(1-p_f)^\ell\leq\frac{e}{2e-3}. \tag{2.5}$$

   We therefore get the first term of R_3 to be bounded by a finite number.

Let F denote the event that in epoch ℓ all the users have unique IDs. This means that in some epoch $\ell_f \in 0, 1, ..., \ell - 1$, the system was fixed. Thus

$$\begin{aligned} & P_\ell(B) \\ & = P(|\hat{\hat{\mu}}_j(m) - \mu_j(m)| > \Delta | F) \\ & = \sum_{i=1}^{\ell-1} P(|\hat{\hat{\mu}}_j(m) - \mu_j(m)| > \Delta | F, \ell_f = i) P(\ell_f = i) \\ & \leq \sum_{i=1}^{\ell-1} 2e^{-2\Delta^2\gamma(\ell-i)} p_f (1-p_f)^{i-1} \\ & = 2e^{-2\Delta^2\gamma\ell} \sum_{i=1}^{\ell-1} e^{2\Delta^2\gamma i} p_f (1-p_f)^{i-1}. \end{aligned} \tag{2.6}$$

Choosing $T_f = M \log(20K)$ gives $p_f > \frac{3(e-1)}{2e}$, and using $\gamma = \frac{1}{2\Delta^2}$ yields

$$P_\ell(B) \leq \frac{4e}{e-1} e^{-\ell} \tag{2.7}$$

and

$$\begin{aligned} \sum_{\ell=1}^{L_T} 2^\ell P_\ell(B) & \leq \frac{4e}{e-1} \sum_{\ell=1}^{L_T} \left(\frac{2}{e}\right)^{-\ell} \\ & \leq \frac{8e}{(e-1)(e-2)}. \end{aligned} \tag{2.8}$$

Thus

$$R_3 \leq C = \frac{e}{2e-3} + \frac{8e}{(e-1)(e-2)}. \tag{2.9}$$

Therefore

$$
\begin{aligned}
R(T) &= R_1 + R_2 + R_3 \\
&\leq \left(\frac{M}{2\Delta^2} + \frac{KM^3}{\log M} \log \frac{1}{\Delta} + 4M^3 \right) \log T + C \qquad (2.10) \\
&= O(\log T).
\end{aligned}
$$

□

2.4 Experimental Results

In this section, we present experimental results to validate the performance of our algorithm. We applied the proposed algorithm to a system with $K = 10$ users and $M = 10$ channels. Figure 2.1 shows the plot of average accumulated regret across the time horizon. The algorithm was run for 10 epochs. We see from Figure 2.1 that the average accumulated regret grows sub-linearly with time and the regret is bounded by $\log T$.

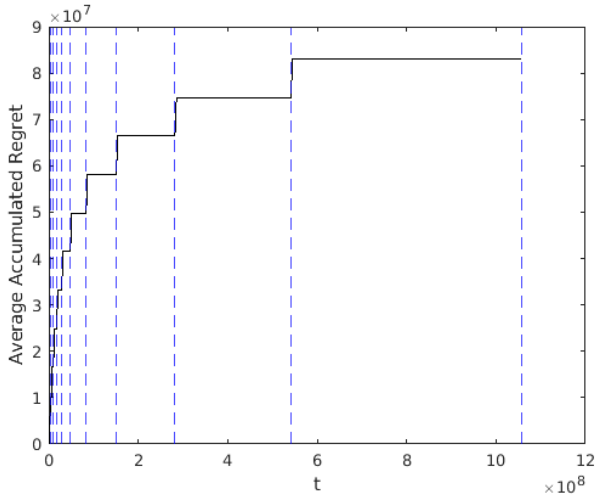

Figure 2.1: Average accumulated regret vs. time

CHAPTER 3

NON-ZERO REWARDS ON COLLISIONS

3.1 System Setting

Consider the set of agents/players $[K] = \{1, 2, \ldots, K\}$. The action space of each player j is the set of M arms $\mathcal{A}_j = [M]$. Let the time horizon be denoted by T, and the action taken by player j at time $t \leq T$ be $a_{t,j}$. The strategy or action profile $\mathbf{a}_t$ is defined as the vector of the actions taken by the players, *i.e.*, $\mathbf{a}_t = [a_{t,1}, ..., a_{t,K}]$. At any given time, the players can observe only their own rewards and cannot observe the actions taken by the other players.

In this model, we assume the reward distribution of each arm to have support $[0, 1]$. In the event that more than one player accesses the same arm m, they could get non-zero rewards. Let $k(a_{t,j})$ denote the number of players playing arm $a_{t,j}$ (including player j). Note that the number of players on arm $a_{t,j}$ is a function of the complete action profile $\mathbf{a}_t$. The reward received by player j pulling arm m, that is pulled by $k(m)$ players, is denoted by $r_j(m, k(m))$. The reward is drawn from a distribution with mean $\mu_j(m, k(m)) = \mathbb{E}[r_j(m, k(m))]$. The mean reward of an arm $\mu_j(m, k(m))$ is inversely proportional to the number of players pulling the arm ($k(m)$). We assume that $\mu_j(m, k(m))$ becomes negligible for some $k(m) \geq N + 1$ where N depends on the system, *i.e.*, $\mu_j(m, k(m)) = 0$ for $k(m) \geq N+1$. When we assume zero rewards on collisions, it is equivalent to assuming that $N = 1$.

The action space $\mathcal{A}$ of the players is simply the product space of the individual action spaces, *i.e.*, $\mathcal{A} = \Pi_{j=1}^{K} \mathcal{A}_j$. Let $\mathbf{a}^* \in \mathcal{A}$ be such that

$$\mathbf{a}^* \in \arg\max_{\mathbf{a} \in \mathcal{A}} \sum_{j=1}^{K} \mu_j(a_j, k(a_j)).$$

In this work, we consider the case where there is a unique optimal matching $\mathbf{a}^*$. Let $J_1 = \sum_{j=1}^{K} \mu_j(a_j^*, k(a_j^*))$ be the system reward for the optimal

matching, and J_2 the system reward for the second optimal matching. In our algorithm we assume that we have access to a lower bound on the value

$$\Delta = \frac{J_1 - J_2}{2MN}$$

Such an assumption is usually required for the analysis of multi-player MABs when communication between players is not allowed [19, 12]. The expected regret during a time horizon T is defined as:

$$R(T) = T \sum_{j=1}^{K} \mu_j(a_j^*, k(a_j^*)) - \mathbb{E}\left[\sum_{t=1}^{T}\sum_{j=1}^{K} \mu_j(a_{t,j}, k(a_{t,j}))\right]$$

where the expectation is over the actions of the players.

For a given arm m, each player j needs to estimate the mean reward $\mu_j(m, n)$ for all $n \in [N]$. In the case where $N = 1$ (zero rewards on collisions), player j estimates $\mu_j(m, 1)$ using all the non-zero rewards obtained. However, when $N \geq 2$, estimating $\mu_j(m, n)$ for $n \in [N]$ is not very straightforward. In order to achieve this and provide some guarantees on the estimation performance, it is necessary to impose a separability condition on the mean rewards as a function of n for any $j \in [K]$ and $m \in [M]$. We use the following separability condition derived from Definition 1 of [23]:

$$|\mu_j(m, n_1) - \mu_j(m, n_2)| \geq 4c\sqrt{\sigma^2 + \epsilon_2}$$

where $j \in [K]$, $m \in [M]$, $n_1, n_2 \in [\beta]$, $\epsilon_2 \in (0, 1)$, σ^2 is the maximum variance of the reward distributions across arms and c is a constant dependent on K and M. A detailed analysis on how we arrive at this condition is provided in Appendix A. This gives us a lower bound on the parameter

$$\nu_{\min} = \min_{j,m,n_1,n_2} |\mu_j(m, n_1) - \mu_j(m, n_2)|$$

where $n_1, n_2 \in [N]$, $\mu_j(m, n_1), \mu_j(m, n_2) \neq 0$, $j \in [K]$ and $m \in [M]$.

3.2 Algorithm

Our proposed policy for each player j in the decentralized multi-player multi-armed bandit setting with non-zero rewards on collisions is presented in Algorithm 3. The policy for a player depends only on the player's own actions and the observed rewards. Our algorithm proceeds in epochs since we do not assume knowledge of time horizon T. Let L_T denote the number of epochs in time horizon T. Each epoch ℓ has three phases: Exploration, Matching and Exploitation.

The exploration phase is for player j to obtain estimates of mean rewards (denoted by $\hat{\mu}_j(m,n)$) of arms $m \in [M]$ for all $n \in [N]$. This phase proceeds for T_0 (expression given in Section 3.3) time units in every epoch. At each time instant t, player j chooses an arm m uniformly from the M arms and stores the reward received. At the end of T_0 time units, player j runs a clustering algorithm (Algorithm 5) on the rewards received while playing arm m, for each m separately. We know that the mean rewards $\mu_j(m,n)$ from $n = 1$ to $n = N$ are decreasing in n. The estimates $\hat{\mu}_j(m,n)$, $n = 1, \ldots, N$ are calculated as the centroids of the clusters sorted in descending order.

Once the players have estimates of the mean rewards on the arms, they need to come to a consensus on the optimal action profile that maximizes the system reward. This is done in the matching phase of each epoch. The parameter ϵ is provided as an input to the algorithm. Given an action profile $\mathbf{a}$, we define the utility of player j to be

$$u_j(\mathbf{a}) = \hat{\mu}_j(a_j, k(a_j)).$$

Section 3.4 explains in detail the matching phase and the choice of the above defined utility function. The matching phase of our proposed algorithm is based on [22], where a decentralized algorithm that leads to maximizing the sum of utilities of the players is presented. This phase proceeds for $c_2 c_\epsilon \ell^{1+\delta}$ time units in epoch ℓ, where c_2 and c_ϵ are constants. At the end of this phase, the players identify the optimal action profile with high probability.

The optimal action profile identified at the end of the matching phase is played in the exploitation phase for $c_3 2^\ell$ time units. As ℓ increases, the players get better estimates of the mean rewards and the probability of identifying the optimal action profile increases. Therefore, the length of the exploitation

phase is set to be exponential in ℓ.

Algorithm 3: Policy for player j

Initialization: Set $\hat{\mu}_j(m,n) = 0$ for all $j \in [K]$, $m \in [M]$ and $n \in [N]$. Initialize empty arrays $r_{j,m}$ for $m \in [M]$. Let L_T be the last epoch with time horizon T

for epoch $\ell = 1$ **to** L_T **do**

Exploration phase:

for $t = 1$ **to** T_0 **do**

Choose an arm m uniformly

Append reward to $r_{j,m}$

end for

Obtain $\hat{\mu}_j(m,n)$ for $n \in [N]$ by running Cluster (Algorithm 3) on $r_{j,m}$ for each m

Matching phase: Run Algorithm 2. Starting from $d = \lceil 0.5 c_2 c_\epsilon \ell^{1+\delta} \rceil$-th turn of Algorithm 2, count the number of times each action was played that resulted in being content:

$$W^\ell(j,m) = \sum_{h=d}^{c_2 c_\epsilon \ell^{1+\delta}} \mathbf{1}_{\{(a_{h,j}=m, S_{h,j}=C)\}}$$

Exploitation phase: For $c_3 2^\ell$ time units, play

$$a_j = \arg\max_m W^\ell(j,m)$$

end for

Theorem 2. *Given the system model specified in Section 3.1, the regret of the proposed algorithm for a time-horizon T and some $0 < \delta < 1$ is $R(T) = O(\log^{2+\delta} T)$.*

Proof. Let L_T be the last epoch with time-horizon T. The regret incurred during the L_T epochs can be analyzed as the sum of the regret incurred in the three phases of the algorithm. From the structure of the algorithm, we can easily see that $L_T < \log T$. Let R_1, R_2 and R_3 denote the regret incurred during the exploration phase, matching phase and exploitation phase during L_T epochs, respectively.

1. Exploration phase: Since the exploration phase in each epoch ℓ proceeds for T_0 time units,

$$R_1 \leq T_0 L_T \leq T_0 \log(T). \tag{3.2}$$

Algorithm 4: Matching phase algorithm

Initialization: Let $c > MN$. $Z_1 = [\bar{a}_{1,j}, \bar{u}_{1,j}, S_{1,j}]$, where $\bar{a}_{1,j} \overset{\text{unif}}{\sim} [M]$, $\bar{u}_{1,j} = 0$ and $S_{1,j} = D$

for $h = 1$ **to** $c_2 \ell^{1+\delta}$ **do**

Action dynamics:

If $S_{h,j} = C$, action

$$a_j = \begin{cases} \bar{a}_{h,j} & \text{with prob } 1 - \epsilon^c \\ a \in [M] \setminus \bar{a}_{h,j} & \text{with uniform prob } \frac{\epsilon^c}{M-1} \end{cases}$$

If $S_{h,j} = D$, action a_j is chosen uniformly from $[M]$

Estimate utility: Upon choosing action a_j, play it for c_ϵ time units and let sample mean of the rewards observed during this duration be $\bar{r}(a_j)$. If $\bar{r}(a_j) = 0$, $\hat{u}_j = 0$. Else let

$$\hat{k}(a_j) = \underset{n \in [N], \hat{\mu}_j(a_j, n) \neq 0}{\arg\min} |\bar{r}(a_j) - \hat{\mu}_j(a_j, n)|$$

The utility of the player is calculated as:

$$\hat{u}_j = \hat{\mu}_j(a_j, \hat{k}(a_j))$$

State Dynamics:

If $S_{h,j} = C$ and $[a_j, \hat{u}_j] = [\bar{a}_{h,j}, \bar{u}_{h,j}]$:

$$Z_{h+1} = Z_h$$

If $S_{h,j} = C$ and $[a_j, \hat{u}_j] \neq [\bar{a}_{h,j}, \bar{u}_{h,j}]$ or $S_{h,j} = D$, the new state

$$Z_{h+1} = \begin{cases} [a_j, \hat{u}_j, C] & \text{with prob } \epsilon^{1-\hat{u}_j} \\ [a_j, \hat{u}_j, D] & \text{with prob } 1 - \epsilon^{1-\hat{u}_j} \end{cases} \tag{3.1}$$

end for

Algorithm 5: Cluster

Input: X

Use an α approximation algorithm to select an initial set of N centroids $C_0 = \{\nu_1, \ldots \nu_N\}$

while Lloyd's algorithm (k-means) has not converged **do**

Run k-means step update

end while

Return centroids of clusters

2. Matching phase: In epoch ℓ, the matching phase runs for $c_2 c_\epsilon \ell^{1+\delta}$ time units. Thus

$$R_2 = \sum_{\ell=1}^{L_T} c_2 c_\epsilon \ell^{1+\delta} \leq c_2 c_\epsilon L_T^{2+\delta} \leq c_2 c_\epsilon \log^{2+\delta} T. \tag{3.3}$$

3. Exploitation phase: In the exploitation phase, regret is incurred in the following two events:

   (a) Let E_1 denote the event that for some player j, arm m and $n \in [N]$, we have that $|\mu_j(m,n) - \hat{\mu}_j(m,n)| \geq \Delta$. Let probability of this event be $P(E_1)$.

   (b) Let E_2^ℓ denote the event that given that all players have $|\mu_j(m,n) - \hat{\mu}_j(m,n)| \leq \Delta$, the action profile chosen in the matching phase of epoch ℓ is not optimal. Let probability of this event for some epoch ℓ be $P(E_2^\ell)$.

   We have from Lemma 1 that by choosing an appropriately large T_0, we have that $P(E_1) \leq c_{\exp} e^{-\ell}$, and from Lemma 4 that $P^(E_2^\ell) \leq C_\rho \exp(-\ell^{1+\delta})$.

   Therefore

$$\begin{aligned} R_3 &= \sum_{\ell=1}^{L_T} c_3 2^\ell (c_{\exp} e^{-\ell} + C_\rho e^{-\ell^{1+\delta}}) \\ &\leq \frac{2c_3(c_{\exp} + C_\rho)}{2+e}. \end{aligned} \tag{3.4}$$

Thus

$$\begin{aligned} R(T) &= R_1 + R_2 + R_3 \\ &\leq T_0 \log T + c_2 c_\epsilon \log^{2+\delta} T + \frac{2c_3(c_{\exp} + C_\rho)}{2+e} \\ &= O(\log^{2+\delta} T). \end{aligned} \tag{3.5}$$

□

3.3 Exploration Phase

The exploration phase is for player j to obtain estimates of mean rewards of arms $m \in [M]$ for all $n \in [N]$. The players explore the arms uniformly and the estimated mean rewards for each arm m are calculated separately by running a clustering algorithm (Algorithm-3) on the rewards obtained while pulling arm m. The estimated rewards $\hat{\mu}_j(m,n)$ for $n = 1$ to N are calculated as the centroids of the clusters sorted in descending order. Algorithm 3 is derived from Algorithm 1 of [23].

Lemma 1. *Given Δ as defined in Section 3.1, for any fixed player j, arm m, and number of players on the arm $n \leq N$, the estimates of the mean rewards $\hat{\mu}_j(m,n)$ obtained after the exploration phase of epoch ℓ for the choice of $T_0 \geq \lceil \frac{32M^K}{(M-1)^{(K-N)}\Delta^2} \rceil$, satisfy*

$$P(|\hat{\mu}_j(m,n) - \mu_j(m,n)| \geq \Delta) \leq c_{\exp} e^{-\ell} \tag{3.6}$$

where $c_{\exp}$ is a constant that depends only on M, K, N.

Proof of this lemma is provided in the Appendix A and closely follows the proof of Theorem 2 of [23] and Theorem 2 of [21]. For a given player j and arm m, we first prove a lower bound on the number of samples obtained from each cluster. We then use the results from [23] along with the separability condition and appropriate concentration inequalities to prove the lemma.

3.4 Matching Phase

Strategic form games in game theory are used to model situations where players choose actions simultaneously and do not have knowledge of the actions of other players. In such games, each player has a utility function $u_j : \mathcal{A} \to [0,1]$, that assigns a real valued payoff to each action profile $\mathbf{a} \in \mathcal{A}$. An algorithm that works under the assumption that every agent can observe only their own action and utility received is called a pay-off based method. The matching phase of our proposed algorithm is based on [22], where a pay-off based decentralized algorithm that leads to maximizing the sum of utilities of the players is presented. In order to pose the multi-player multi-armed bandit problem as a strategic form game, we need to design the utility

functions of the players in a way such that the system regret is minimized. Consider the utility of player j associated with the action profile $\mathbf{a}$ to be denoted by $u_j(\mathbf{a})$. We define the utility as:

$$u_j(\mathbf{a}) = \hat{\mu}_j(a_j, k(a_j)).$$

The action profile that maximizes the sum of utilities is called an efficient action profile. We have from Lemma 1 of [19] that if

$$|\hat{\mu}_j(m,n) - \mu_j(m,n)| \leq \Delta \tag{3.7}$$

for all $j \in [K]$, $m \in [M]$, $n \in [N]$, and for Δ as defined in Section 3.1, then we have that

$$\arg\max_{\mathbf{a}\in\mathcal{A}} \sum_{i=1}^{K} \mu_j(a_j, k(a_j)) = \arg\max_{\mathbf{a}\in\mathcal{A}} \sum_{i=1}^{K} \hat{\mu}_j(a_j, k(a_j)).$$

While Lemma 1 of [19] provides a proof under the assumption of zero rewards on collisions, a similar proof can be worked out for the case of non-zero rewards on collisions. The condition given by (3.7) is guaranteed with high probability by the exploration phase of the algorithm. Thus, the efficient action profile that maximizes the sum of utilities (estimated mean rewards) is the same as the optimal action profile that minimizes regret or, equivalently, maximizes system performance.

3.4.1 Description of algorithm

This phase proceeds in 'plays', where each play lasts c_ϵ time units and ϵ is a parameter of the algorithm. Each player j is associated with a state $Z_h = [\bar{a}_{h,j}, \bar{u}_{h,j}, S_{h,j}]$ during play h, where $\bar{a}_{h,j} \in [M]$ is the baseline action of the player, $\bar{u}_{h,j} \in [0,1]$ is the baseline utility of the player and $S_{h,j} \in \{C, D\}$ is the mood of the player (C denotes "content" and D denotes "discontent").

When the player is content, the baseline action is chosen with high probability $(1-\epsilon^c)$ and every other action is chosen with uniform probability. If the player is discontent, the action is chosen uniformly from all arms and there is a high probability that the player would choose an arm different from the baseline action. This part of the algorithm constitutes the action dynamics.

The baseline action can be interpreted as the arm the agent expects to play in the efficient action profile and the baseline utility can be interpreted as the pay-off the player expects to receive upon playing the baseline action. The player being content is an indication that the payoff received by the player by playing his baseline action is satisfactory and as expected. Thus, the goal in designing the algorithm is for all the players to align the baseline actions and baseline utilities to the efficient action profile and be content in this state.

We have seen the justification for using the utility function $u_j(\mathbf{a}) = \hat{\mu}_j(a_j, k(a_j))$ in the introduction of Section 3.4. However, at each time instant in the matching phase, the player receives only the instantaneous reward and does not know $k(a_{t,j})$ corresponding to the action profile chosen in the previous step, in order to determine the utility. Thus, we estimate the $k(a_{t,j})$ as $\hat{k}(a_j)$ and $u_j(\mathbf{a})$ as $\hat{u}_j(\mathbf{a})$. This is done by the player pulling the arm chosen in the action dynamics for c_ϵ time units and recording the sample mean of the rewards observed during this duration as $\bar{r}_j(a_j)$. The estimate $\hat{k}(a_j)$ is given by:

$$\hat{k}(a_j) = \underset{n \in [N], \hat{\mu}_j(a_j,n) \neq 0}{\arg\min} |\bar{r}(a_j) - \hat{\mu}_j(a_j, n)|$$

and $\hat{u}_j(\mathbf{a}) = \hat{\mu}_j(a_j, \hat{k}(a_j))$.

Lemma 2. *If* $c_\epsilon \geq \lceil \frac{2\ln(\frac{2}{\epsilon^c})}{(\Delta + 2\nu_{min})^2} \rceil$, *we have that*

$$p_\epsilon = P(u_j(\mathbf{a}) \neq \hat{u}_j(\mathbf{a})) \leq \epsilon^c.$$

The proof follows directly from Hoeffding's inequality. Thus, we have an estimate of the utility of the player that is correct with high probability. Note that in Algorithm 4, $\hat{u}_j(\mathbf{a})$ is referred to as just $\hat{u}_j$ for readability.

The player updates his current state by comparing the action played and the estimate of the utility received with his baseline action and baseline utility. If the player is content and his baseline action and utility match the action played and the estimate of the utility, the state remains the same. Otherwise, the next state is chosen probabilistically based on the estimate of the utility. The rationale behind the particular probabilities chosen is that when the utility received is high, the player is more probable to be content.

The utility each player receives is equivalent to feedback from the system on how the entire action profile affects the reward received by this player. If the player receives a lower payoff due to that arm not being good or due to

collisions, there is a higher probability of the player becoming discontent and exploring other arms. On the other hand, if the payoff received is higher, there is a higher probability of the player staying content and exploiting the same arm again. Thus the agent dynamics and state dynamics balance the exploration exploitation trade-off in the multi-player bandit setting.

From the $d = c_2 c_\epsilon \ell^{1+\delta}$-th play of the matching phase algorithm, the player counts the number of times each arm was played that resulted in being content:

$$\sum_{h=d}^{c_2 c_\epsilon \ell^{1+\delta}} \mathbf{1}_{\{(a_{h,j}=m, S_{h,j}=C)\}}.$$

The action chosen by the player for the exploitation phase is

$$a_j = \arg\max_m W^\ell(j, m).$$

3.4.2 Analysis of the matching phase algorithm

The matching phase algorithm is based on the work in [22], and the guarantees provided there state that the action profile achieving the optimal sum of utilities is played for a majority of the time. The analysis of the algorithm in [22] relies on the theory of regular perturbed Markov decision processes [24].

The dynamics of the matching phase algorithm induce a Markov chain over the state space $\mathcal{Z} = \Pi_{j=1}^K([M] \times [0,1] \times \mathcal{M})$ where $\mathcal{M} = \{C, D\}$. Let P^0 denote the probability transition matrix of the process when $\epsilon = 0$ and P^ϵ denote the transition matrix when $\epsilon > 0$. The process P^ϵ is a regular perturbed Markov process if for any $z, z' \in \mathcal{Z}$ (Equations (6),(7) and (8) of Appendix of [24]):

1. P^ϵ is ergodic
2. $\lim_{\epsilon \to 0} P^\epsilon_{zz'} = P^0_{zz'}$
3. $P^\epsilon_{zz'} > 0$ implies for some ϵ, there exists $r \geq 0$ such that $0 < \lim_{\epsilon \to 0} \epsilon^{-r} P^\epsilon_{zz'} < \infty$

The value of r satisfying the third condition is called the resistance of the transition $z \to z'$, denoted by $r(z \to z')$.

Let μ^ϵ be the unique stationary distribution of P^ϵ, where P^ϵ is a regular perturbed Markov process. Then $\lim_{\epsilon\to 0}\mu^\epsilon$ exists and the limiting distribution μ^0 is a stationary distribution of P^0. The stochastically stable states are the support of μ^0. The main result of [22] (Theorem 1) states that the stochastically stable states maximize the sum of utilities of the players. Our matching phase algorithm differs from [22] in two aspects. We prove that, in spite of those differences, the conditions required in their paper to prove their result are still satisfied, and thus the stochastically stable states of our matching phase algorithm maximize the sum of the utilities of the players.

It can be easily verified that the algorithm presented in [22] satisfies the conditions for regular perturbed Markov process. The transitions where $P^\epsilon_{zz'} > 0$ and $\lim_{\epsilon\to 0} P^\epsilon_{zz'} = 0$ are called ϵ perturbations. The first way in which our algorithm differs from the one in [22] is that the players in our algorithm do not directly observe their utilities. Instead they estimate their utilities, and there is a probability of error of p_ϵ in this step. However, this can be represented as an ϵ perturbation by rewriting the state update step as follows (for readability, the index of the play h is dropped in the following equations):

If $S_j = C$:

If $[a_j, \hat{u}_j] = [\bar{a}_j, \bar{u}_j]$, the new state is

$$[\bar{a}_j, \bar{u}_j, C] \to \begin{cases} [\bar{a}_j, \bar{u}_j, C] & \text{w.p.} \quad 1 - p_\epsilon \\ [\bar{a}_j, u_j, C], & \text{w.p.} \quad p_\epsilon(\epsilon^{1-u_j}) \\ [\bar{a}_j, u_j, D], & \text{w.p.} \quad p_\epsilon(1-\epsilon^{1-u_j}). \end{cases} \tag{3.8}$$

If $[a_j, \hat{u}_j] \neq [\bar{a}_j, \bar{u}_j]$, with $q < 1$

$$[\bar{a}_j, \bar{u}_j, C] \to \begin{cases} [\bar{a}_j, \bar{u}_j, C] & \text{w.p.} \quad qp_\epsilon \\ [a_j, u_j, C], & \text{w.p.} \quad (1-qp_\epsilon)(\epsilon^{1-u_j}) \\ [a_j, u_j, D], & \text{w.p.} \quad (1-qp_\epsilon)(1-\epsilon^{1-u_j}). \end{cases} \tag{3.9}$$

If $S_j = D$:

$$[\bar{a}_j, \bar{u}_j, D] \to \begin{cases} [a_j, u_j, C] & \text{w.p. } (1-p_\epsilon)\epsilon^{1-u_j} \\ [a_j, u_j, D], & \text{w.p. } (1-p_\epsilon)(1-\epsilon^{1-u_j}) \\ [a_j, \hat{u}_j, C], & \text{w.p. } p_\epsilon \epsilon^{1-\hat{u}_j} \\ [a_j, \hat{u}_j, D], & \text{w.p. } p_\epsilon(1-\epsilon^{1-\hat{u}_j}). \end{cases} \tag{3.10}$$

We have from Lemma 2 that $p_\epsilon \leq \epsilon^c$. Thus we can see that the unperturbed process is the same as in [22] and it can be easily verified that these dynamics satisfy the conditions for a regular perturbed Markov chain specified in [24].

The second way in which our dynamics differ from [22] is that our strategic form game is not interdependent (Definition 1 of [22]). The interdependence property implies that it is not possible to divide the agents into two distinct subsets, where the actions of agents in one subset do not affect the utilities of those in the other. However, in our case, consider an action profile where $N+1$ players play an arm m. These $N+1$ players will receive zero utility, no matter what the other players play. Thus, our game is not interdependent. The only time this property is used in [22] is to find the recurrence classes of P^0. We can prove that our dynamics produce the same recurrence classes as in [22] using the structure of our algorithm.

Lemma 3. *Let D^0 represent the set of states in which everyone is discontent. Let C^0 represent the set of states in which each agent is content and the benchmark action and utility are aligned. Then the recurrence classes of the unperturbed process are D^0 and all singletons $z \in C^0$.*

The proof is provided in Appendix B.

Let D be any state in D^0 and $z, z' \in C^0$. It can be seen that the resistances for the paths $z \to D$, $D \to z$ and $z \to z'$ in our algorithm are the same as in [22]. For instance, the transition from $z \to D$ occurs only when a player explores or the utility is miscalculated. Since we have $p_\epsilon \leq \epsilon^c$, the probability of this event is $O(\epsilon^c)$ and hence the resistance of the transition is c. Similarly it can be seen that the resistance for the path $D \to z$ is $(K - \sum_{j\in[K]} \bar{u}_j)$ and $z \to z'$ is bounded in $[c.2c)$.

Thus, we can use the proof of Theorem 1 of [22] to say that the stochastically stable states of our matching phase algorithm maximize the sum of utilities. A more detailed explanation of the same is provided in Appendix B. Since we assume a unique optimal action profile, the state with the base-

line actions and utilities corresponding to the optimal action profile and all players being content is the stochastically stable state. Since this state is played for a majority of the time, we can bound the probability of the optimal action profile not being identified at the end of the matching phase as in the following result.

Lemma 4. *In some epoch ℓ, let $\mathbf{a}^* = \arg\max_{\mathbf{a}\in\mathcal{A}} \sum_{j=1}^{K} u_j(\mathbf{a})$ and let $\mathbf{a}' = [a'_1, ..., a'_K]$ where $a'_j = \arg\max_{m\in[M]} W^\ell(j,m)$ at some epoch ℓ. For small enough ϵ,*

$$P(\mathbf{a}^* \neq \mathbf{a}') \leq C_\rho \exp\left(-\ell^{1+\delta}\right)$$

for some $C_\rho > 0$.

The proof relies on using Chernoff-Hoeffding bounds for Markov chains (Theorem 3 of [25]) and is provided in Appendix B.

3.5 Simulation Results

In this section, we present the results of simulations carried out to validate the performance of our algorithm. We consider two cases, one with $K = M$ and the other with $K > M$. In both cases, we allow for maximum of 2 players on each arm ($N = 2$). The mean rewards $\mu_j(m, 1)$ for player j, arm m are generated uniformly at random from $[0.3, 0.95]$. The mean rewards $\mu_j(m, 2)$ are generated as $\mu_j(m, 2) = 0.5\mu_j(m, 1) + u$, where u is a uniform random variable in $[-0.05, 0.05]$. The rewards are generated from a uniform distribution with variance 0.003. In our simulations, we set $\delta = 0$. The values of mean rewards for each case is provided in Appendix C.

Due to numerical considerations, we modify the probabilities in (3.1) as $\epsilon^{u_j^{\max}-\hat{u}_j}$ and $1 - \epsilon^{u_j^{\max}-\hat{u}_j}$ instead of $\epsilon^{1-\hat{u}_j}$ and $1 - \epsilon^{1-\hat{u}_j}$, where $u_j^{\max}$ is the maximum utility that can be received by the player.

To the best of our knowledge, the setting we have considered that allows for heterogeneous reward distributions and non-zero rewards on collisions has not been studied prior to this work. Thus, it is not possible to compare our algorithms against existing algorithms.

3.5.1 $K = M$

We consider a system with $K = 6$ players and $M = 6$ arms. The optimal action profile is $\mathbf{a}^* = [2\ 1\ 1\ 6\ 4\ 5]$. The mean reward for arm 3 is small for all the players and thus the optimal allocation favors arm 1 for players 2 and 3. The considered system has $\Delta = 0.05$ and $\nu_{\min} = 0.1110$. We set $T_0 = 4.1 \times 10^5$, $c_2 = c_3 = 10^4$ and $c_\epsilon = 1.1 \times 10^4$. The value of ϵ is set to 10^{-5}. Since it is possible for each player to play a distinct arm and receive non-zero rewards, $u_j^{\max} = \max_{m \in [M]} \hat{\mu}_j(m, 1)$ for all $j \in [K]$. The algorithm was run for 10 epochs and the experiment was repeated for 85 iterations and the accumulated regret averaged over the iterations.

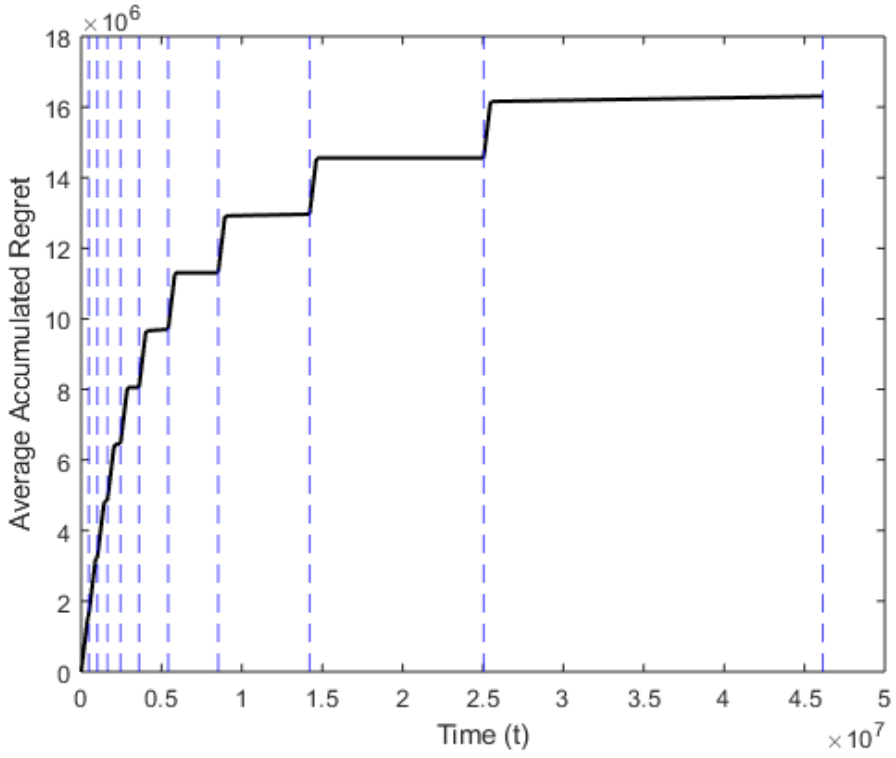

Figure 3.1: Average accumulated regret as a function of time

3.5.2 $K > M$

We consider a system with $K = 6$ players and $M = 3$ arms. The optimal action profile is $\mathbf{a}^* = [2\ 3\ 2\ 1\ 1\ 3]$. The considered system has $\Delta = 0.0397$ and $\nu_{\min} = 0.0757$. We set $T_0 = 7.4 \times 10^5$, $c_2 = c_3 = 2 \times 10^4$ and $c_\epsilon = 1.1 \times 10^4$.

The value of ϵ is set to 10^{-4}. Since it is required for 2 players to choose each arm for all the players to receive non-zero rewards, $u_j^{\max} = \max_{m \in [M]} \hat{\mu}_j(m, 2)$. The algorithm was run for 10 epochs and the experiment was repeated for 100 iterations and the accumulated regret averaged over the iterations.

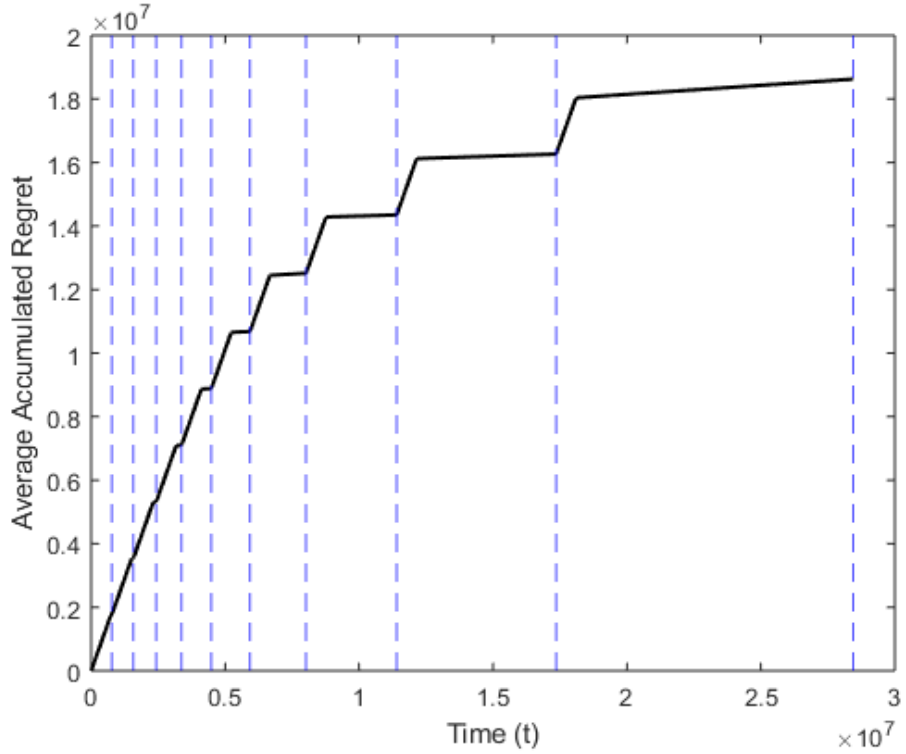

Figure 3.2: Average accumulated regret as a function of time

For the purpose of simulations in the case of $K > M$, we assume that the number of players is known. When the number of players is not known, we can estimate the number of players using methods similar to Algorithm 1 of [14]. Additional details are provided in Appendix C.

From Figure 3.1 and Figure 3.2, we see that the average accumulated regret grows sub-linearly with time. We can also observe that the average regret incurred during the exploitation phase of each epoch is small as the matching phase converges to the optimal action profile with high probability.

CHAPTER 4

CONCLUSION

In this work, we have studied the uncoordinated spectrum access problem using a multi-player multi-armed bandit framework where there is no central control and users cannot communicate with each other. Under the considered model, the reward distributions for the same channel may vary across the users. Using this model, we have considered two settings - zero and non-zero rewards on collisions. Under the zero rewards on collision model, we present an algorithm that achieves average accumulated regret of order $O(\log T)$ for time horizon T, which is order optimal. In the second part of our work, we consider the more practical non-zero rewards on collision model. Under that setting, we present a policy that achieves average accumulated regret of $O(\log^{2+\delta} T)$ for time horizon T, for some $0 < \delta < 1$. The policy presented for the non-zero rewards on collision setting is aimed towards small networks and may be applied to bigger networks by division into subsystems. The policy presented for the zero rewards on collision scenario is easily applicable to large networks.

APPENDIX A

EXPLORATION PHASE

The exploration phase of the algorithm is for player j to obtain estimates of mean rewards for the arms $m \in [M]$ and for all $n \in [N]$. The estimated mean rewards for each arm m are calculated separately by running a clustering algorithm (Algorithm 3) on the rewards obtained while pulling arm m. The estimated rewards $\hat{\mu}_j(m,n)$, for $n = 1$ to N, are calculated as the centroids of the clusters sorted in descending order. The separability condition on the mean rewards of arm m as a function of n (number of players choosing that arm) provided in Section 3.1 is required to provide guarantees on the clustering performance.

In section A.1, we provide some notation and the idea behind the proof of Lemma 1. In section A.2, we provide an insight for the particular choice of the separability condition. A proof sketch for Lemma 1 is provided in subsection A.3.

A.1 Clustering

For readability, let $\mathcal{R} = \{r_1, \ldots, r_T\}$ denote the set of rewards received by some player $j \in [K]$ for some arm $m \in [M]$ ($r_{j,m}$ in Algorithm 1). The clustering algorithm (Algorithm 3) is run on $\mathcal{R}$ to obtain N cluster centers corresponding to the estimated rewards $\hat{\mu}_j(m,n)$ for $n = 1$ to N. Let $\{\mathcal{T}_n\}_{n\in[N]}$ denote the true partition of $\mathcal{R}$, *i.e.*, if $r_t \in \mathcal{T}_n$, then r_t is drawn from the reward distribution corresponding to player j, arm m, and n number of players on the arm, with mean reward $\mu_j(m,n)$. For ease of notation, μ_n and $\mu_j(m,n)$ are used interchangeably in this section. Let $t_n = |\mathcal{T}_n|$. Let $\{\mathcal{S}_n\}_{n\in[N]}$ denote the clusters given by Algorithm 3. Let $g(\mathcal{S}) = \frac{1}{|\mathcal{S}|}\sum_{r\in\mathcal{S}} r$ denote the mean of the elements of the set $\mathcal{S}$. Let $\phi(\mu_n, \mathcal{S}_n) = \sum_{r\in\mathcal{S}_n} |r-\mu_n|^2$. The cost of the clustering algorithm with respect to the true mean rewards

is defined as $\Phi = \sum_{n=1}^{N} \phi(\mu_n, \mathcal{S}_n)$.

The guarantees for the clustering algorithm provided in [23] (Lemma 5) involve using a separability condition and a cost function (Section 1 of [23]) with respect to the sample means of the true partition ($g(\mathcal{T}_n)$ for $n \in [N]$). In our work, we use an approach similar to [23] to provide high probability guarantees with respect to the true means of the distributions corresponding to the clusters $\{\mu_n\}_{n\in[N]}$.

A.2 Separability Condition

We assume the following separability condition (stated in Section 3.1) on the mean rewards of the arms:

$$|\mu_j(m, n_1) - \mu_j(m, n_2)| \geq 4c'\sqrt{\sigma^2 + \epsilon_2} \tag{A.1}$$

for all $j \in [K]$, $m \in [M]$, $n_1, n_2 \in [N]$, $\epsilon_2 \in (0, 1)$, where σ^2 is the maximum variance of the reward distributions across arms and c' is a constant dependent on K and M. In order to get to a separability condition similar to the one in Definition 1 of [23], we first provide a lower bound on the number of samples observed by player j during the exploration phase, drawn from the reward distribution corresponding to arm m, with n players on the arm. The following lemma is adapted from Lemma 4 of [26].

Lemma 9. *If $T_0 = \lceil \frac{32M^K}{(M-1)^{(K-N)}\Delta^2} \rceil$, then each player j observes at least $t_n = \frac{16}{\Delta^2} \log \frac{2MKN(N+1)}{\delta}$ samples drawn from the reward distribution corresponding to arm m, with n players on the arm, during each epoch ℓ.*

Combining the above lemma with (A.1), we obtain a separability condition similar to the one in Definition 1 of [23].

Lemma 10. *For any player j and arm m, if the separability condition* (A.1) *is satisfied, and for all $n \in [N]$ $t_n \geq \frac{16}{\Delta^2} \log \frac{2MKN(N+1)}{\delta}$, then for any $n_1, n_2 \in [N]$, with probability greater than $1 - \frac{\delta}{2MKN(N+1)}$, we have that*

$$|\mu_{n_1} - \mu_{n_2}| \geq c \left[\frac{1}{\sqrt{t_{n_1}}} + \frac{1}{\sqrt{t_{n_2}}} \right] \sqrt{\Phi} \tag{A.2}$$

where $c > 16$ is a constant.

The proof follows directly from Hoeffding's inequality.

A.3 Proof Sketch of Lemma 1

We use techniques similar to those used in Section 3 of [23] to provide guarantees for the performance of the clustering algorithm. Let $\Delta_n = |\mu_n - g(\mathcal{S}_n)|$ and $\gamma = \max_{n_1,n_2 \neq n_1 \in [N]} \frac{\Delta_{n_1}}{|\mu_{n_1} - \mu_{n_2}|}$. Let $\rho_{in}^n = \frac{\sum_{p \neq n} |\mathcal{T}_p \cap \mathcal{S}_n|}{t_n}$ and $\rho_{out}^n = \frac{\sum_{p \neq n} |\mathcal{T}_n \cap \mathcal{S}_p|}{t_n}$, *i.e.*, $\rho_{in}^n + \rho_{out}^n$ captures the fraction of misclassifications in $\mathcal{S}_n$ with respect to $\mathcal{T}_n$. We first present the following lemma that is analogous to Lemma 2 of [23] that is useful in proving the required result.

Lemma 11. *If $\gamma < \frac{1}{4}$, the following hold for all $r \in \mathcal{S}_n$ for all $n \in [N]$*

$$|r - \mu_p| \geq (\frac{1}{2} - 2\gamma)|\mu_n - \mu_p|, \quad \forall p \neq n \tag{A.3}$$

$$|r - \mu_p| \leq \frac{1}{1 - 4\gamma}|r - \mu_n|. \tag{A.4}$$

We now present the following lemmas that are high probability versions of Lemma 1 and Lemma 3 of [23] respectively.

Lemma 12. *If* (A.2) *is satisfied and an α-approximation algorithm returns the set of centroids $\{g(\mathcal{S}_1, \ldots, g(\mathcal{S}_\mathcal{N})\}$, then we have that $\forall \mathcal{T}_n$, $\exists p$ such that $|g(S_n) - \mu_p| \leq \sqrt{2\alpha + 2}\frac{\sqrt{\Phi}}{t_n}$ and $\gamma \leq \frac{2(\alpha+1)}{c}$ with probability $1 - \kappa_1 e^{-\kappa_2 \ell}$ for epoch ℓ (where κ_1, $\kappa_2 > 0$).*

Lemma 13. *If $\gamma < \frac{1}{4}$ and* (A.2) *is satisfied, then $\rho_{in}^n \leq \frac{2}{(1-4\gamma)^2 c^2}$ and $\rho_{out}^n \leq \frac{2}{(1-4\gamma)^2 c^2}$ with probability $1 - \kappa_3 e^{-\kappa_4 \ell}$ for epoch ℓ (where κ_3, $\kappa_4 > 0$).*

The proofs of the above lemmas use the techniques used in Lemma 1 and Lemma 3 of [23] along with Hoeffding's inequality.

The following lemma is analogous to Lemma 5 of [23], and the proof follows directly by using the proofs of Lemma 4 and Lemma 5 of [23].

Lemma 14. *Given that the results of Lemma 7 and Lemma 9 hold, we have that*

$$|g(\mathcal{S}_n) - \mu_n| \leq 2\left[\sqrt{\frac{\rho_{out}^n}{t_n}} + 2\sqrt{\frac{\rho_{in}^n}{t_n}}\right]\sqrt{\Phi}. \tag{A.5}$$

Note that Φ is the sum of independent and bounded random variables. Thus we can use Hoeffding's inequality to upper bound Φ with high probability. Combining this with Lemma 9, we arrive at the required result in Lemma 1.

APPENDIX B

MATCHING PHASE

The matching phase algorithm is based on the work in [22], and the guarantees provided there state that the action profile achieving the optimal sum of utilities is played for a majority of the time. This is stated in Theorem 1 of [22] and the proof relies on the theory of resistance trees for regular perturbed Markov processes [24].

Theorem 1 of [22] states that for an interdependent game on a finite action space, the stochastically stable states of the dynamics presented in their paper maximize the sum of utilities of all players. By definition of the stochastically stable state, this state is played for a majority of time eventually. Our goal is to use the proof of Theorem 1 of [22] to show that the stochastically stable state of the matching phase dynamics maximizes the sum of utilities of these players.

Our matching phase algorithm differs from the dynamics presented in [22] in two aspects. Despite these two differences, we show that the proof technique of Theorem 1 of [22] can be adapted to our dynamics as well.

The first way in which our algorithm differs from that in [22] is that the players in our algorithm do not directly observe their utilities. Instead they estimate their utilities, and there is a probability of error of p_ϵ in this step. We have shown in Section 3.4 that despite this difference, our matching phase dynamics satisfies the conditions for a regular perturbed Markov chain. Therefore, we can use the theory of regular perturbed Markov chains [24] to analyze our algorithm.

The second way in which our dynamics differ from [22] is that our strategic form game is not interdependent (Definition 1 of [22]). However, the only place the interdependence property is used in the proof of Theorem 1 of [22] is to characterize the recurrence classes of the unperturbed Markov process P^0. We prove in the Lemma 3 that even though the interdependence property does not hold for our dynamics, the recurrence classes of P^0 are the same as

those obtained in Lemma 2 of [22].

Thus, we can use the proof of Theorem 1 of [22] to say that the stochastically stable states of our matching phase algorithm maximize the sum of utilities. Since we assume a unique optimal action profile, the state with the baseline actions and utilities corresponding to the optimal action profile and all players being content is the stochastically stable state. We bound the probability of the optimal action profile not being identified at the end of the matching phase in Lemma 4.

B.1 Proof of Lemma 3

In the unperturbed process, if all the players are discontent, they remain discontent with probability 1. Thus we have that D^0 represents a single recurrence class. In each state $z \in C^0$, each player chooses their baseline action, and since the utilities received would be the same as the baseline utilities, each player stays content with probability 1. Thus we have that D^0 and the singletons $z \in C^0$ are recurrence classes.

To see that the above are the only recurrence classes, look at any state that has at least one discontent player and at least one content player. We have that the baseline actions and utilities of the content players are aligned (since this is the unperturbed process). Consider one of the discontent players. This player chooses an action at random and there is a positive probability (bounded away from 0) of choosing the action of a content player. This would cause the utility of the content player to become misaligned with his baseline utility, thus leading to that player becoming discontent. This continues until all players become discontent. Thus any such state cannot be a recurrent state.

Now consider a state where all agents are content, but there is at least one player j whose benchmark action and utility are not aligned. For the unperturbed process, in the following step, the same action profile would be played but this would cause player j to become discontent and it follows from the previous argument that this leads to all players becoming discontent.

Thus we have that D^0 and all singletons in C^0 are the only recurrent states of the unperturbed process.

B.2 Proof of Lemma 4

Let $\bar{\mathbf{u}}$ denote the utilities of the players for the optimal action profile $\mathbf{a}^*$. The optimal state of the Markov chain is then $\mathbf{z}^* = [\mathbf{a}^*, \bar{\mathbf{u}}, C^K]$. Let the distribution of the Markov chain after the first $d = \lceil 0.5c_2\ell^{1+\delta} \rceil$ plays be ϕ. When the system is in state z, let the observed state as seen by the players be z'. The observed state would differ from the true state only in the utilities, due to the possibility that some players may calculate their utilities incorrectly. The counting process in the matching phase of some epoch ℓ is done for $L = \lfloor 0.5c_2\ell^{(1+\delta)} \rfloor$ plays. In order to bound the probability of the event $\{\mathbf{a}^* \neq \mathbf{a}'\}$, we use the Chernoff-Hoeffding bounds for Markov chains from [25], which is also used in [19]. The function $f(z)$ considered here is

$$f(z) = \mathbf{1}_{\{z=z^*, z'=z\}}.$$

This is the event when the current state is the optimal state and the state is also observed correctly (*i.e.* the utilities are estimated correctly by all players). If the optimal state is played for more than $L/2$ plays and the utilities are calculated correctly by all the players for these plays, then the optimal action profile would be played in the exploitation phase. Thus we have that

$$\begin{aligned} P\{\mathbf{a}^* \neq \mathbf{a}'\} &\leq P\left\{\sum_{\tau=1}^{L} f(z(\tau)) \leq L/2\right\} \\ &= P\left\{\sum_{\tau=1}^{L} \mathbf{1}_{\{z(\tau)=z^*(\tau), z'(\tau)=z(\tau)\}} \leq L/2\right\}. \end{aligned} \tag{B.1}$$

In order to use the bounds in [25], we need $\mathbb{E}[f(z)]$, which can be computed as

$$\begin{aligned} \mathbb{E}[f(z)] &= P\{z = z^*, z' = z\} \\ &= P\{z = z^*\} P\{z' = z | z = z^*\} \end{aligned} \tag{B.2}$$

We also have that

$$
\begin{aligned}
P\{z' \neq z | z = z^*\} &= P\left\{\cup_{j\in[K]} \hat{u}_j \neq \bar{u}_j\right\} \\
&\leq \sum_{j\in[K]} P\{\hat{u}_j \neq \bar{u}_j\} \\
&\leq K\epsilon^c.
\end{aligned} \tag{B.3}
$$

from Lemma 2.

This gives $\mathbb{E}[f(z)] = \mu_f \geq P\{z = z^*\}(1 - K\epsilon^c)$. And from the definition of a stochastically stable state, we can choose an ϵ small enough such that $P\{z = z^*\}(1 - K\epsilon^c) > p > 1/2$. Define $\eta = 1 - \frac{1}{2\mu_f}$ so that $0 < \eta < 1$ when $\mu_f > 1/2$. Let T be the 1/8 mixing time of the Markov chain. Then from Theorem 3 of [25] we have that

$$
\begin{aligned}
&P\left\{\sum_{\tau=1}^{L} f(z(\tau)) \leq L/2\right\} \\
&= P\left\{\sum_{\tau=1}^{L} f(z(\tau)) \leq (1-\eta)\mu_f L\right\} \\
&\leq c_0 ||\phi||_\pi \exp(-\frac{(1-\frac{1}{2\mu_f})^2 \mu_f c_2 \ell^{1+\delta}}{144T}) \\
&\leq C_\rho \exp\left(-\ell^{1+\delta}\right)
\end{aligned} \tag{B.4}
$$

where $\rho = \frac{(1-\frac{1}{2\mu_f})^2 \mu_f c_2 (1-\rho)}{72T} > 0$.

APPENDIX C

DETAILS ON SIMULATIONS

Due to numerical considerations, we modify the state update step in the matching phase of the algorithm. The state update step of the matching phase algorithm is as follows:
If $S_{h,j} = C$ and $[a_j, \hat{u}_j] = [\bar{a}_{h,j}, \bar{u}_{h,j}]$:

$$Z_{h+1} = Z_h.$$

If $S_{h,j} = C$ and $[a_j, \hat{u}_j] \neq [\bar{a}_{h,j}, \bar{u}_{h,j}]$ or $S_{h,j} = D$, the new state

$$Z_{h+1} = \begin{cases} [a_j, \hat{u}_j, C] & \text{with prob } \epsilon^{1-\hat{u}_j} \\ [a_j, \hat{u}_j, D] & \text{with prob } 1 - \epsilon^{1-\hat{u}_j}. \end{cases} \tag{C.1}$$

We modify the probabilities in (C.1) as $\epsilon^{u_j^{\max} - \hat{u}_j}$ and $1 - \epsilon^{u_j^{\max} - \hat{u}_j}$, instead of $\epsilon^{1-\hat{u}_j}$ and $1 - \epsilon^{1-\hat{u}_j}$ respectively. Here $u_j^{\max}$ denotes the maximum utility that can be received by player j, without reducing the utility of any other player to zero. For example, when there are $K = 6$ players and $M = 6$ arms, each player can occupy a separate arm and all the players could receive non-zero utilities. Thus the value of $u_j^{\max}$ of player j would be $u_j^{\max} = \max_{m \in [M]} \hat{\mu}_j(m, 1)$. Consider the case when there are $K = 6$ players, $M = 3$ arms and $N = 2$. In this case, it is not possible for any player j to occupy an arm by himself without reducing the utilities of some other players to zero. In this case, $u_j^{\max} = \max_{m \in [M]} \hat{\mu}_j(m, 2)$.

For the purpose of simulations in the case of $K > M$, we assume that the number of players in the system is known in order to calculate $u_j^{\max}$ for each player j. If the number of players is not known, we can adapt the method from the 'Musical Chairs' algorithm in [14] to estimate the number of players. Note that in the musical chairs algorithm, each player knows the number of collisions (the event where one or more other players also

choose the same arm) that occur and this is used to estimate the number of players in the system. However, in our setting, players cannot observe the number of collisions as we allow for non-zero rewards on collisions. Instead, we can provide a separate phase in the beginning of the algorithm for the each player to estimate the total number of players in the system. This can be done by each player choosing an arm uniformly, and playing the chosen arm for a certain number of time units to estimate the number of players on that arm. This way, the number of collisions that occur during this phase can be obtained with high probability. This is similar to the method used to estimate the utilities of the players in the matching phase of the algorithm.

REFERENCES

[1] W. R. Thompson, "On the likelihood that one unknown probability exceeds another in view of the evidence of two samples," *Biometrika*, vol. 25, no. 3/4, pp. 285–294, 1933.

[2] T. L. Lai and H. Robbins, "Asymptotically efficient adaptive allocation rules," *Advances in Applied Mathematics*, vol. 6, no. 1, pp. 4–22, 1985.

[3] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite-time analysis of the multiarmed bandit problem," *Machine Learning*, vol. 47, no. 2-3, pp. 235–256, 2002.

[4] A. Garivier and O. Cappé, "The kl-ucb algorithm for bounded stochastic bandits and beyond," in *Proceedings of the 24th Annual Conference on Learning Theory*, 2011, pp. 359–376.

[5] S. Bubeck, N. Cesa-Bianchi et al., "Regret analysis of stochastic and nonstochastic multi-armed bandit problems," *Foundations and Trends in Machine Learning*, vol. 5, no. 1, pp. 1–122, 2012.

[6] T. Lattimore and C. Szepesvári, *Bandit Algorithms*. Cambridge University Press, 2020.

[7] N. Cesa-Bianchi, C. Gentile, Y. Mansour, and A. Minora, "Delay and cooperation in nonstochastic bandits," in *Conference on Learning Theory*, vol. 49, 2016, pp. 605–622.

[8] B. Szörényi, R. Busa-Fekete, I. Hegedűs, R. Ormándi, M. Jelasity, and B. Kégl, "Gossip-based distributed stochastic bandit algorithms," in *Journal of Machine Learning Research Workshop and Conference Proceedings*, vol. 2. International Machine Learning Society, 2013, pp. 1056–1064.

[9] N. Korda, B. Szörényi, and L. Shuai, "Distributed clustering of linear bandits in peer to peer networks," in *Journal of Machine Learning Research Workshop and Conference Proceedings*, vol. 48. International Machine Learning Societ, 2016, pp. 1301–1309.

[10] N. Evirgen and A. Kose, "The effect of communication on noncooperative multiplayer multi-armed bandit problems," in *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE, 2017, pp. 331–336.

[11] O. Avner and S. Mannor, "Multi-user lax communications: a multi-armed bandit approach," in *IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications*. IEEE, 2016, pp. 1–9.

[12] D. Kalathil, N. Nayyar, and R. Jain, "Decentralized learning for multiplayer multiarmed bandits," *IEEE Transactions on Information Theory*, vol. 60, no. 4, pp. 2331–2345, 2014.

[13] O. Avner and S. Mannor, "Concurrent bandits and cognitive radio networks," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2014, pp. 66–81.

[14] J. Rosenski, O. Shamir, and L. Szlak, "Multi-player bandits–a musical chairs approach," in *International Conference on Machine Learning*, 2016, pp. 155–163.

[15] A. Anandkumar, N. Michael, A. K. Tang, and A. Swami, "Distributed algorithms for learning and cognitive medium access with logarithmic regret," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 4, pp. 731–745, 2011.

[16] E. Boursier and V. Perchet, "SIC-MMAB: synchronisation involves communication in multiplayer multi-armed bandits," in *Advances in Neural Information Processing Systems*, 2019, pp. 12 048–12 057.

[17] L. Besson and E. Kaufmann, "Multi-player bandits revisited," *arXiv preprint arXiv:1711.02317*, 2017.

[18] H. Tibrewal, S. Patchala, M. K. Hanawal, and S. J. Darak, "Distributed learning and optimal assignment in multiplayer heterogeneous networks," in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*. IEEE, 2019, pp. 1693–1701.

[19] I. Bistritz and A. Leshem, "Distributed multi-player bandits-a game of thrones approach," in *Advances in Neural Information Processing Systems*, 2018, pp. 7222–7232.

[20] E. Boursier, V. Perchet, E. Kaufmann, and A. Mehrabian, "A practical algorithm for multiplayer bandits when arm means vary among players," *arXiv preprint arXiv:1902.01239*, 2019.

[21] M. Bande and V. V. Veeravalli, "Multi-user multi-armed bandits for uncoordinated spectrum access," in *2019 International Conference on Computing, Networking and Communications (ICNC)*. IEEE, 2019, pp. 653–657.

[22] J. R. Marden, H. P. Young, and L. Y. Pao, "Achieving pareto optimality through distributed learning," *SIAM Journal on Control and Optimization*, vol. 52, no. 5, pp. 2753–2770, 2014.

[23] C. Tang and C. Monteleoni, "On Lloyd's algorithm: New theoretical insights for clustering in practice," in *Artificial Intelligence and Statistics*, 2016, pp. 1280–1289.

[24] H. P. Young, "The evolution of conventions," *Econometrica: Journal of the Econometric Society*, pp. 57–84, 1993.

[25] K.-M. Chung, H. Lam, Z. Liu, and M. Mitzenmacher, "Chernoff-Hoeffding bounds for Markov chains: Generalized and simplified," *arXiv preprint arXiv:1201.0559*, 2012.

[26] M. Bande and V. V. Veeravalli, "Multi-user multi-armed bandits for uncoordinated spectrum access," *arXiv e-prints*, p. arXiv:1807.00867, Jul 2018.